Cycle Calendar Generator Documentation

Release 1.0.0

Ryan William Maynard Oldford

Apr 25, 2020

Contents:

Cycle Calendar Generator is a command line application designed to take schedule data for *n*-day cycles and produce iCal files, ready to import into the calendar software of your choice.

Is your school on a 6-day cycle? 7-day? Cycle Calendar Generator can make iCal files for all your teachers or students!

CHAPTER 1

Quickstart

Assuming you have Python 3.6+ installed, use *pip3 <http://www.pip-installer.org/>*.

```
$ pip3 install cycle_calendar_generator
```

Set up your schedule data Excel files as shown in the *Examples*. Put them all in one folder, and run Cycle Calendar Generator from the command line/Terminal.:

```
$ cycle_calendar_generator path/to/excel/files
```

If you're running the Generator from that folder, no folder input is needed!

```
$ cycle_calendar_generator
```

Find your iCal files in the /output folder (i.e. path/to/excel/files/**output**)

1.1 Cycle Calendar Generator

Generates iCal files for class schedules for schools using an N-day cycle

If you're a teacher or a student, you probably have a 6 or 7-day cycle. You'd like to enter your schedule into your calendar, but calendar software doesn't support "recur every n weekdays". So you're out of luck, right? That's where Cycle Calendar Generator comes in. You just need to make an Excel file with your school's schedule, and another Excel file for each teacher/student's schedule, and Cycle Calendar Generator does the rest.

- Documentation: https://cycle-calendar-generator.readthedocs.io.

1.1.1 Installation

```
$ pip install cycle_calendar_generator
```

1.1.2 Usage

1. **Make a "Schedule Setup" Excel file**
    a. Filename must be schedule_setup.xlsx
    b. Has 3 sheets, named "Period Timing", "Cycle Days List", and "Yearly Schedule"
    c. "Period Timing" sheet gives the period name/number, start time, and end time for each period
    d. "Cycle Days List" sheet lists the name/number of each day in the cycle
    e. "Yearly Schedule" sheet lists all dates in the school year and the matching day in the cycle
    f. Dates and times should be in the standard Excel date/time format
    g. All other data should be in text format, including numbers.
    h. See below for examples
2. **Make an Excel file for each user that wants a schedule calendar**
    a. The filename should match the user's name, and will be used to name the output calendar file. (Ex. Eric Idle's Excel file should be named "Eric Idle.xlsx", and will generate "Eric Idle.ics")
    b. Has 1 sheet, named "User Schedule", that has the user's usual schedule for a cycle
    c. All data here should be in text format
    d. See below for examples
3. Put all files in any folder
4. Run the application as follows:

    ```
    $ cycle_calendar_generator path/to/schedule/files
    ```

5. Schedule iCal files are found in the /output folder under the folder with your input Excel files.

1.1.3 Examples

Period Timing

Period Number	Start Time	End Time
1	8:00	9:00
2	9:00	10:00
3	10:00	11:00
4	11:00	12:00
5	12:00	13:00

Times can be in either 24h or 12h format. "Period Number" should be text format, not number.

Cycle Days List

A1	B2	C3	D4	E5	F6

The entries here are the official names for all cycle days. Every cycle day entry in other sheets must match these values.

Yearly Schedule

Date	Cycle Day
8/31/18	A1
9/3/18	B2
9/4/18	C3
9/5/18	D4
9/6/18	E5
9/7/18	F6

Dates can be displayed any way, but must be date format. Entries in the "Cycle Day" column must be an official cycle day as defined in Cycle Days List.

User Schedule

Period Number	A1	B2	C3	D4	E5	F6
1	Grade 8		Grade 11			Grade 8
2		Grade 11			Grade 8	
3	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch
4	Grade 11			Grade 8		Grade 11
5			Grade 8		Grade 11	

"Period Number" should be text format, not number. The "Cycle Days" in the top row must be official cycle days as defined in Cycle Days List

1.1.4 Tests

For current Python version: `python3 setup.py test`

For versions 3.5+ `tox`

It's recommended to use pyenv to install Python versions required by tox. pyenv can be installed using Homebrew:

```
brew update
brew install pyenv
pyenv install 3.6.6 3.7.6
```

1.1.5 Contribute

Contributions are always welcome! For thoughts on features or bug reports see Issues. If you're interested in contributing to this library, see details on doing so in the CONTRIBUTING.rst file in this repository.

1.1.6 Credits

This package was created with Cookiecutter and the elgertam/cookiecutter-pipenv project template, based on audreyr/cookiecutter-pypackage.

Excel file reading was made possible by the openpyxl package, while iCal file reading and writing uses the ics package.

[~ Dependencies scanned by PyUp.io ~]

1.1.7 Licence

- Free software: GNU General Public License v3

1.2 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

1.2.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/ROldford/cycle_calendar_generator/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

Cycle Calendar Generator could always use more documentation, whether as part of the official Cycle Calendar Generator docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/ROldford/cycle_calendar_generator/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

1.2.2 Get Started!

Ready to contribute? Here's how to set up *cycle_calendar_generator* for local development.

1. Fork the *cycle_calendar_generator* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/cycle_calendar_generator.git
```

3. Install your local copy using pipenv. Assuming you have pipenv installed, this is how you set up your fork for local development:

```
$ pipenv install --dev
$ pipenv shell
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 cycle_calendar_generator tests
$ python setup.py test or py.test
```

   tox should be used outside of the pipenv shell:

```
$ deactivate
$ tox
$ pipenv shell
```

   flake8 and tox are

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

1.2.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/ROldford/cycle_calendar_generator/pull_requests and make sure that the tests pass for all supported Python versions.

1.2.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_cycle_calendar_generator
```

1.3 History

1.3.1 0.1.0 (2018-08-09)

- First release on PyPI.

1.3.2 0.1.1 - 0.1.8 (2018-10-18 to 10-19)

- Attempts to fix issues with Travis CI's PyPI deployment

1.3.3 0.2.0 (2018-10-19)

- Program can now be installed as a Unix-style command line application

1.3.4 1.0.0 (2020-02-25)

- No longer works on Python 3.5
- Dependency update